Prebid Server and First Party Data

Updated Feb 15, 2023

1. Overview

The First Party Data (FPD) feature set provides a standard way for the clients to supply first party data and control which bidders have access to it.

Objectives:

1. Establish a standard convention for where pages, apps, and dooh requests can place first party data
2. Give the publisher control over which bidders are allowed to see the first party data
3. Adapters should be updated to read from the standard location, mapping values to their bidder-specific locations.
4. Use OpenRTB conventions where possible. This means that OpenRTB standard attributes like domain, keywords, etc are handled differently than arbitrary attributes.
5. Support Prebid.js, Prebid SDK, and AMP
6. Try to utilize as much FPD as possible. FPD will become increasingly important to improve bid values.
7. Prebid Server must support bidder-specific overrides for the following ORTB objects: site, app, user, and dooh for all supported versions of ORTB. It should support bidder-specific overrides for all valid ORTB fields.

1.1. OpenRTB Interface

PBJS and the SDK place values in a number of OpenRTB locations as described in the following table.

OpenRTB Attribute	Description	PBJS Source	SDK Source	AMP Source	PBS Support
ext.prebid.data.bidders[]	If specified, only these bidders are allowed to see fields in {site/app/user/dooh}.ext.data.	n/a	addBidderToAccessControlList()	bidders	Java only so far
site.ATTR	Only standard OpenRTB attributes should be in the root level: name, domain, cat, sectioncat, pagecat, page, ref, search, keywords, etc. Notes: 'site.content.data[]' is where taxonomy data is placed. 'site.ext.data' is where generic FPD is placed.	setConfig({ortb2.site.ATTR});	n/a	site.ATTR	both Java and Go pass to adapters, some fields subject to bidder permissions as defined below.
app.ATTR	Only standard OpenRTB attributes should be in the root level: name, bundle, domain, storeurl, cat, sectioncat, pagecat, keywords, etc. Notes: 'app.content.data[]' is where taxonomy data is placed. 'app.ext.data' is where generic FPD is placed.	setConfig({ortb2});	n/a	n/a	both Java and Go pass to adapters, though some fields subject to bidder permissions
user.ATTR	Only standard OpenRTB attributes should be in the root level: yob, gender, keywords, etc. Note: user.ext.data is where generic FPD is placed.	setConfig({fpd.user.ATTR});	n/a	user.ATTR	both Java and Go pass to adapters

user.data[]	Prebid.js specifications used user.data rather than user.ext.data. Hard to change this now. PBS behavior is: if data is an array, keep it as-is. If data is an object, merge into user.ext.data and remove user.data.	setConfig({fpd.user.data.ATTR});	n/a	user.data	both Java and Go pass to adapters, subject to bidder permissions
dooh.ATTR	Only standard OpenRTB attributes should be in the root level: venuetype, keywords, etc. Notes: dooh.content.data[]' is where taxonomy data is placed. 'dooh.ext.data' is where generic FPD is placed.	n/a	n/a	n/a	TBD
imp[].ext.ATTR	AdUnit-specific attributes go here	AdUnit.ortb2ImpExt	n/a	AMP ATTR goes into imp.ext.data.ATTR	both Java and Go pass to adapters
ext.prebid.bidderconfig.ortb2	Bidder-specific config is merged into the object of record for the specified bidders.	setBidderConfig()	stored request	stored request	both Java and Go pass to adapters

1.2. OpenRTB Examples

Example 1

```
{
    ext: {
       prebid: {
           data: { bidders: [ "bidderA" ] }  // limit bidders that receive global data
       }
    },
    site: {
         keywords: "",
         search: "",
         ext: {
             data: {
                 // only seen by bidderA as named in ext.prebid.data.bidders[]
                 GLOBAL CONTEXT DATA
             }
         }
    },
    user: {
        keywords: "",
        gender: "",
        yob: 1999,
        geo: {},
        ext: {
            data: {
                // only seen by bidderA as named in ext.prebid.data.bidders[]
```

```
                GLOBAL USER DATA }
        }
    },
    imp: [
        ...
        ext: {
                // everyone sees this data
                gpid: "/11111/homepage#uniquifier",
                data: {
                    ADUNIT SPECIFIC CONTEXT DATA
                }
         }
    ]
}
```

Example 2

```
{
    ext: {
       prebid: {
           bidderconfig: [ {
               bidders: [ 'bidderA', 'bidderB' ],
               config: {
                      "ortb2": {
                            "site": {
                                "ext": {
                                    "data": {
                                        "customsite": "customsite1"
                                    }
                                }
                            },
                            "user": {
                                "ext": {
                                    "data": {
                                        "customuser": "customuser1"
                                    }
                                }
                            }
                        }
               }
            },{
               bidders: [ 'bidderC' ],
               config: {
                  ortb2: { site: { ... }, user: { ...} }
               }
            }]
       }
    }
}
```

2. Prebid Server Core FPD Requirements

The job of PBS-Core in relation to OpenRTB First Party Data is:

1) Validate and normalize certain pieces of first party data
2) Control which bidders see first party data attributes
3) In some instances, modify or remove privacy-sensitive data

Here's the algorithm:

All merges should prefer source over destination. e.g. if we're merging a bidder-specific bidderconfig into the ortb2 object and the "site.BLAH" field exists in both, PBS would override site.BLAH with the new value.

1. For the raw request only:
    a. If user.data exists as an object, merge it into user.ext.data and remove user.data. This is to cover older versions of Prebid.js where the specifications defined user.data{}. Hard to change this now.
        i. "Merge" is defined as a deep JSON merge with arrays being replaced instead of appended.
    ~~b. To transition legacy FPD bidder permissions, if ext.prebid.bidderconfig[].config.fpd exists:~~
        ~~i. Merge ext.prebid.bidderconfig[].config.fpd.context.data into ext.prebid.bidderconfig[].config.ortb2.site.ext.data~~
        ~~ii. Merge ext.prebid.bidderconfig[].config.fpd.context into ext.prebid.bidderconfig[].config.ortb2.site.~~
        ~~iii. Merge ext.prebid.bidderconfig[].config.fpd.user.data into ext.prebid.bidderconfig[].config.ortb2.user.ext.data~~
        ~~iv. Merge ext.prebid.bidderconfig[].config.fpd.user into ext.prebid.bidderconfig[].config.ortb2.user.~~
    c. Normalize all fields defined with an "alternative data type" instead of throwing them away
    d. JSON validation
    e. Note: storedrequests are not normalized
2. When creating the bidder-specific request:
    a. Copy the whole OpenRTB JSON for that bidder
    b. If ext.prebid.data.bidders[] exists and this bidder isn't in it, remove site.ext.data.*, app.ext.data.*, dooh.ext.data.*, and user.ext.data.*, and {site,app,dooh}.content.data[] and user.data[].

c. If ext.prebid.bidderconfig exists and this bidder is in scope, merge supplied values (in config.ortb2 only) to the bidder ORTB, but if ext.prebid.data.bidders exists, only for bidders named there.
    i. If the same bidder was seen in more than one ext.prebid.bidderconfig.bidder, reject the request with a string warning: "the same bidder may not have more than one bidderconfig entry."
    ii. Bidderconfig.config.ortb2 entries may contain any valid ORTB field.
    iii. Any value in bidderconfig.config outside of ortb2 should be ignored.
d. If the resulting request has more than one of {app,site,dooh}, reject request for this bidder
e. Copy imp[].ext**.prebid.bidder.**BIDDER to imp[].ext.bidder
f. Remove any imps[] that don't contain this bidder
g. Remove all other imp[].ext.BIDDERs
h. Remove ext.prebid.data.bidders
i. Remove ext.prebid.bidderconfig
j. Privacy controls
    i. Process COPPA, LMT, USP, and GDPR rules -- remove user IDs and round addresses/geo as required.

Note: the /openrtb2/video endpoint is out of scope for First Party Data.

2.1. /openrtb2/auction endpoint

Requirements

1) Attributes should be validated against any defined data types.
2) Where transformations are specified, attributes should be normalized to the defined data types.
3) If an attribute doesn't pass defined validation checks, it should be removed from the request with a warning placed in the messages section of debug output and a log message emitted N% of the time. The auction should continue.
4) When a transformation is done on the data, a warning may be placed in the debug output to encourage originators to adhere to the preferred standard.
5) Attributes defined in ext.prebid.bidderconfig.config.ortb2 should be merged into the OpenRTB object for the bidders named in ext.prebid.bidderconfig.bidders, but if ext.prebid.data.bidders exists, the bidder needs to be in that list. Here's how the merge should work:
    a) If ext.prebid.bidderconfig.config.ortb2 is a valid ORTB field, merge it to the bidder's request.
        i) The definition of merge is: bidder-specific override takes precedence. Arrays are replaced rather than appended.
    b) If the request is now invalid because it contains more than one of site, dooh, and app, reject the request for this bidder.

6) Privacy rules:
    a) Follow COPPA anonymization rules
    b) Follow us_privacy anonymization rules
    c) Follow TCF2 rules
7) If the ext.prebid.data.bidders field is specified, remove the OpenRTB attributes flagged as in-scope for permissions from any bidder (or alias) not present in the array.
    a) Note: PBS should not reject the whole request if there's an unrecognized bidder or alias in this array.
8) The server should be able to handle future changes to OpenRTB relating to the site, app, dooh, and user objects. For instance, in ORTB2.6, new fields such as site.cattax were added. These should be supported at a bidder-specific level.

The following table details all the fields that can come into Prebid Server. Column descriptions:

- **Attribute** - a first party data attribute or pattern of attributes. Important: any attribute flagged as "Requires Permissions" in the last column can come from either the top level OpenRTB **or** ext.prebid.bidderconfig.config.ortb2.
    - Note that {site,app,user}.data isn't mentioned here because they should have been merged into {site,app,user}.ext.data by this point
- **Normalized Data Type** - the expected data type of the attribute, and how it should be passed to bidder adapters.
- **Alternate Data Types** - the web is a messy place. Data comes in unexpected formats. This column defines the data transformations that Prebid Server should do to convert data to the Normalized Data Type.
    - Note: fields in ext.prebid.bidderconfig.ortb2.* should follow the same normalization rules as their destination. E.g. ext.prebid.bidderconfig.ortb2.site.name=[“a”,”b”] would be normalized to site.name=”a” because ‘name’ is one of the special OpenRTB attributes that maps into the site object and there is a normalization rule defined for it.
    - In contrast, ext.prebid.bidderconfig.ortb2.site.id=[“1”,”2”] maps to site.ext.data.id, which doesn’t have any normalization rules defined, so the value is left untouched.
- **Description** - notes about the attribute
- **Requires Permissions** - defines whether an attribute is in-scope for First Party Data permissions, Prebid Server treats it specially:
    - This field will be removed from the bidder copy of the OpenRTB if it's not permitted by ext.prebid.data.bidders.
    - The attributes flagged as 'meta' are always removed before going to a bidder.

Attribute	**Normalized Data Type**	**Alternate Data Types (PBS-Java only)**	**Description**	**In-Scope for Perms?**

{site/app/dooh}.id	string	array of strings. Use first element.	OpenRTB attribute	N
{site/app/dooh}.name	string	array of strings. Use first element.	OpenRTB attribute	N
app.bundle	string	array of strings. Use first element.	OpenRTB attribute	N
app.storeurl	string	array of strings. Use first element.	OpenRTB attribute	N
{site/app/dooh}.domain	string	array of strings. Use first element	OpenRTB attribute	N
{site/app}.cat	array of strings	-	OpenRTB attribute	N
{site/app}.sectioncat	array of strings	-	OpenRTB attribute	N
{site/app}.pagecat	array of strings	-	OpenRTB attribute	N
site.page	string	array of strings. Use first element	OpenRTB attribute	N
site.ref	string	array of strings. Use first element	OpenRTB attribute	N
site.search	string	array of strings. Use first element	OpenRTB attribute	N
{site/app/dooh}.content.ATTR other than 'data'	various	-	OpenRTB attribute	N
{site/app/dooh}.content.data	array	-	OpenRTB attribute	Y
{site/app/dooh}.publisher	object	-	OpenRTB attribute	N
{site/app/dooh}.keywords	string	array of strings. Concatenate array elements with commas	OpenRTB attribute	N

site.mobile	integer	-	OpenRTB attribute	N
{site/app}.privacypolicy	integer	-	OpenRTB attribute	N
site.ext.data.ATTR	any	-	OpenRTB attribute	Y
app.ext.data.ATTR	any	-	OpenRTB attribute	Y
{site/app}.ext.*	n/a	-	OpenRTB attribute	N
{site/app}.ATTR	any	-	OpenRTB attribute	N
user.keywords	string	array of strings. Concatenate array elements with commas	OpenRTB attribute	N
user.gender	string	array of strings. Use first element	OpenRTB attribute	N
user.yob	integer	-	OpenRTB attribute	N
user.geo	object	-	OpenRTB attribute	N
user.data	object	-	OpenRTB attribute	Y
user.ext.data.ATTR	any	-	OpenRTB attribute	Y
user.ATTR	any	-	OpenRTB attribute	N
imp[].ext.ATTR	any	-	Arbitrary attributes	N
ext.prebid.data.bidders	array of strings	-	If present, instructs Prebid Server to remove first party data fields from global sections for some bidders.	meta
ext.prebid.bidderconfig	object	-	If present, instructs Prebid Server to inject the named data into the ORTB object for the named bidders.	meta

2.2. /openrtb2/amp endpoint

Assumptions

- The AMP "TGT" macro is resolved from json.targeting. Because these values are also utilized by GAM, there cannot be any additional structure imposed on this object.
- Dynamic values (e.g. user IDs or segments) are not supported in AMP. When they are, we will revise this spec.
- We could add additional fields to pass static data through the AMP call, but it's easier to just store them in the storedrequest database.
- No one started making use of the original FPD implementation. No need for backwards compatibility.

Requirements

1) The AMP openrtb in the stored request must be able to carry global site data.
2) The stored request must also be able to define bidder permissions either with ext.prebid.data.bidders or ext.prebid.bidderconfig.
    a) It's ok if both kinds of permission are specified: i.e. every entry in ext.prebid.bidderconfig[].bidders would also need to be in ext.prebid.data.bidders or it will be ignored.
3) Some fields defined in the stored request will be overridden by values coming in on the query string:
    a) curl overwrites site.page
    b) account overwrites site.publisher.id if it's valid
4) Read the 'targeting' attribute on the AMP query string. Add all key-value-pairs found there to imp[].ext.data.
5) PBS must adhere to privacy rules as for other interfaces:
    a) Follow COPPA anonymization rules
    b) Follow usp_privacy anonymization rules
    c) Follow TCF2 rules

Example

```
<amp-ad width="300" height="50"
  type="doubleclick"
  data-slot="/1111/amp_test"
  data-multi-size-validation="false"
  rtc-config='{"vendors": {"prebidrubicon": {"REQUEST_ID": "blah"}}}'
  json='{ "targeting": {"gam-key1": "val1","gam-key2": "val2"}}' >
</amp-ad>
```

GET
/openrtb2/amp?tag_id=blah&w=300&h=250&ow=&oh=&ms=&slot=%2F1111%2Famp_test&**targeting=%7B%22gam-key1%22%3A%22val1%22%2C%22gam-key2%22%3A%22val2%22%7D**&...

OpenRTB results

```
{
  ...
  "imp": [
    ...
    "ext": {
       "data": {
           "gam-key1": "val1",
           "gam-key2": "val2"
       }
    }
  ]
}
```

Appendix - Change Log

Date	Person	Update
Jul 11, 2022	bretg	Added Req 10: The server should be able to handle future changes to OpenRTB. For instance, in ORTB2.6, new fields such as cattax and wlangb were added. These should all be supported at a bidder-specific level.
Oct 11, 2022	bretg	By request: allowing bidder-specific config to contain valid ORTB fields other than site/app/user. Removed previous changes and struck-out redundant passages.
Feb 15, 2023	bretg	Added DOOH.